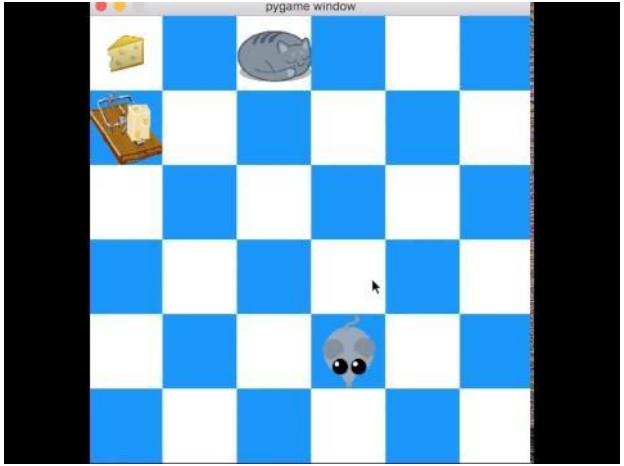
Reinforcement Learning

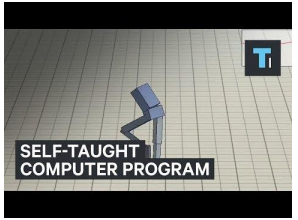
Q learning

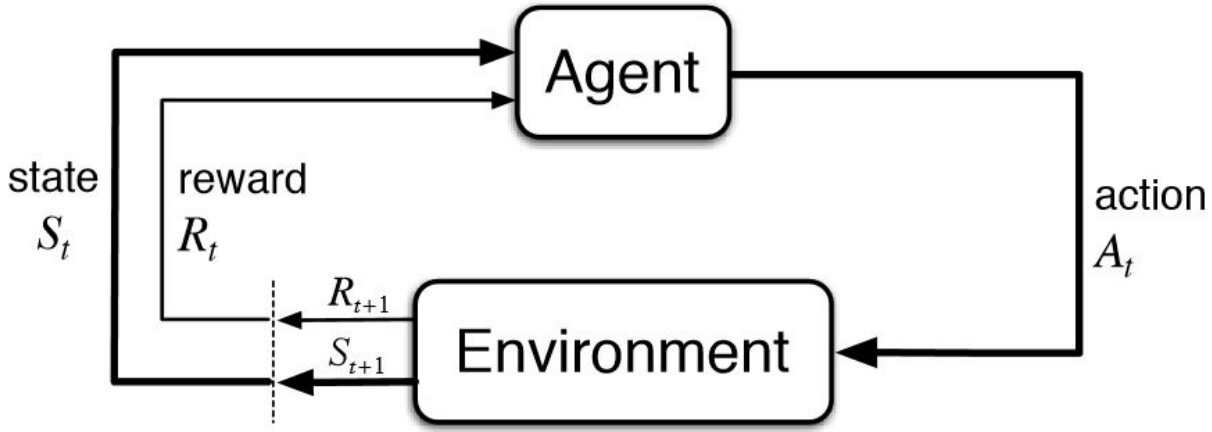


D

What is Reinforcement Learning?

- ➔ It is a way for a machine to learn how to optimally behave in its environment constrained by its own goals
- ➔ Markov Decision Process: Framework for modeling decision in a non/semi stochastic environment





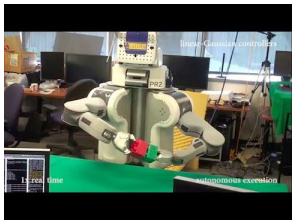
Why Reinforcement Learning

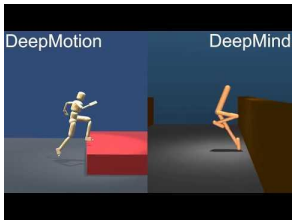
- ➔ Learning beyond data
  - ◆ Supervised/Unsupervised are only as good as the data they are fed
- ➔ Learning beyond humans
  - ◆ Machine can never be better than human
- ➔ Learning complex optimal policies
  - ◆ Replacing human intuition and rule based policies
- ➔ Complex applications
  - ◆ Robotics, Autonomous vehicles, Games

Current Applications

Autonomous Vehicles

Robotics





Games

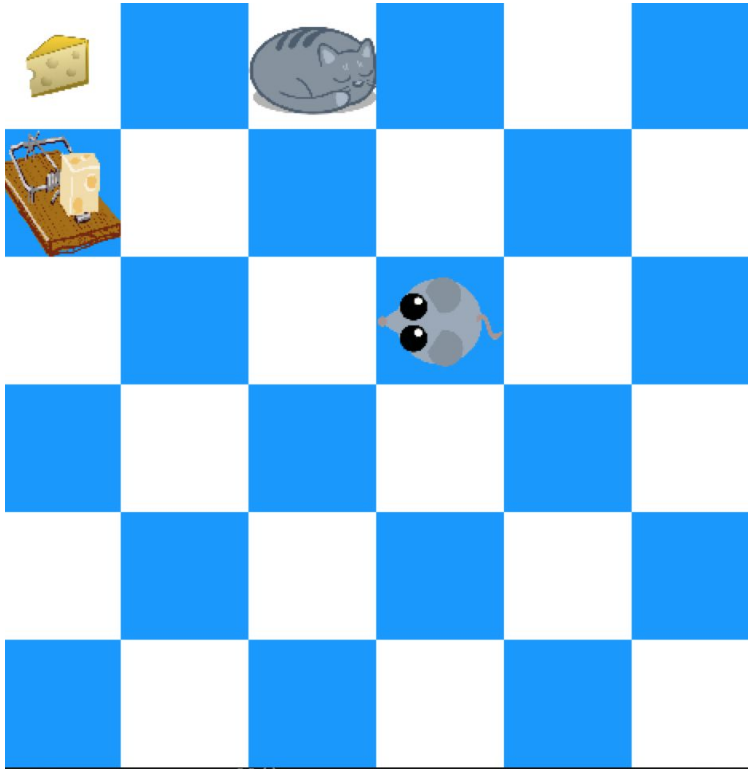
How does it work?

- ➔ Environment
  - ◆ Agents world with physical and non physical components
- ➔ State
  - ◆ Snapshot of the environments (Physical/Non-Physical) at a given moment
- ➔ Action
  - ◆ Agent interaction with the environment
- ➔ Rewards
  - ◆ Feedback from the environment
- ➔ Q Table
  - ◆ Values we want to learn

References: https://leonardoaraujosantos.gitbooks.iol

Our game

- ➔ Environment: Agents, walls, fixtures, etc
- ➔ States: (Position) * (Mouse, Cheese, Trap, Cat)
  - ◆ Number of states
- ➔ Actions: Left, Right, Up, Down
- ➔ Rewards: (Cheese = 1) (Cat/Trap = -1)
- ➔ Parameters: Q[(state, action)] = Val
  - ◆ Number of values



Learning

- ➔ Optimized Policy: Mapping between actions and states
- ➔ Q-Learning (Based on Bellman equation for optimal policy)

$$NewQ(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma max Q'(s',a') - Q(s,a)]$$

New Q value for that state and that action

Current Q value

Learning Rate

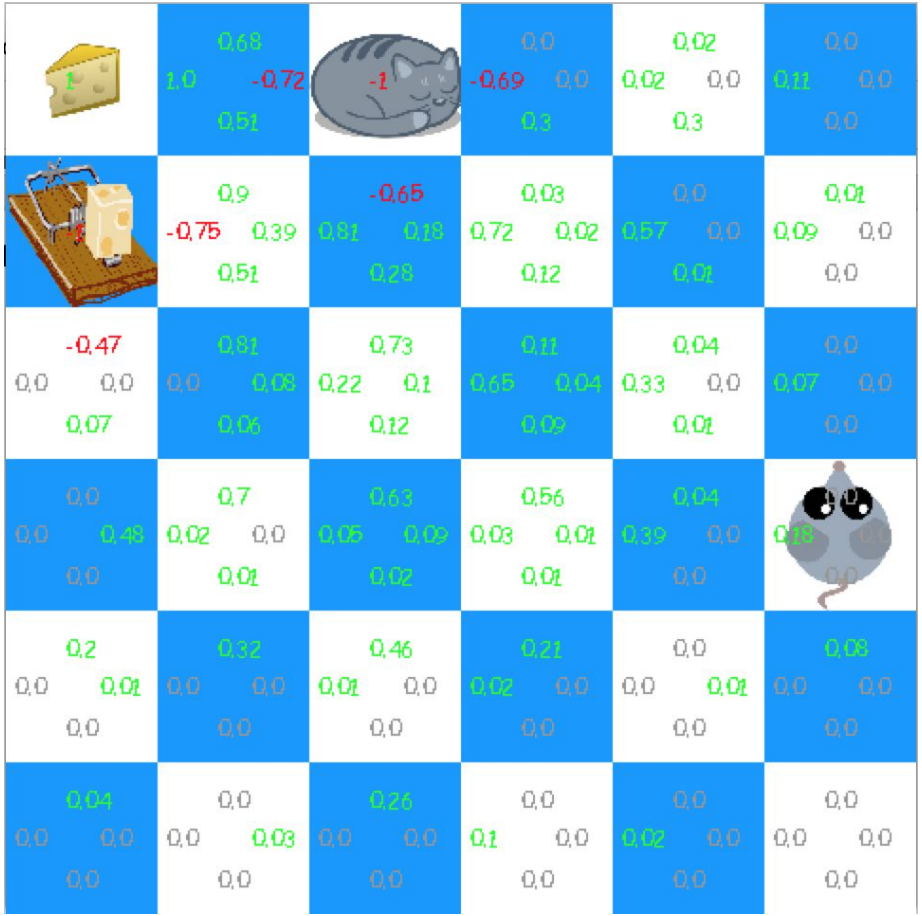
Reward for taking that action at that state

Discount rate

Maximum expected future reward **given the new s' and all possible actions at that new state**

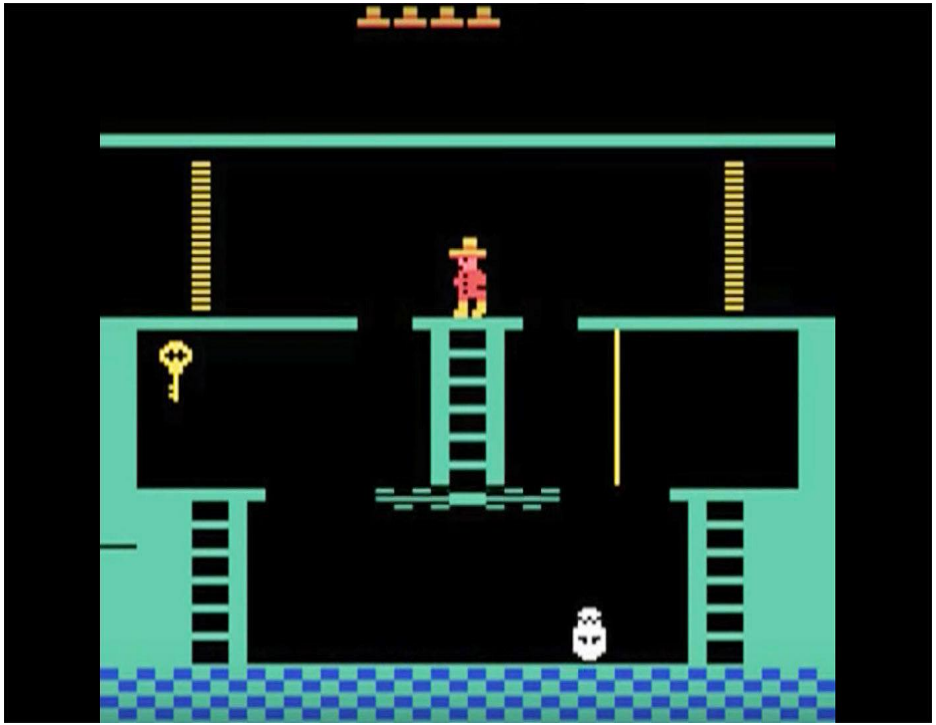
- ➔ Discount Factor: Allow us to reduce the importance of early steps
- ➔ Epsilon: Exploration vs Exploitation

Learning


1
0,68
1,0 -0,72
0,51
-1
0,0
-0,69 0,0
0,3
0,02
0,02 0,0
0,3
0,0
0,11 0,0
0,0
-1
0,9
-0,75 0,39
0,51
-0,65
0,81 0,18
0,28
0,03
0,72 0,02
0,12
0,0
0,57 0,0
0,01
0,01
0,09 0,0
0,0
-0,47
0,0 0,0
0,07
0,81
0,0 0,08
0,06
0,73
0,22 0,1
0,12
0,11
0,65 0,04
0,09
0,04
0,33 0,0
0,01
0,0
0,07 0,0
0,0
0,0
0,0 0,48
0,0
0,7
0,02 0,0
0,01
0,63
0,05 0,09
0,02
0,56
0,03 0,01
0,01
0,04
0,39 0,0
0,0
0,2
0,0 0,01
0,0
0,32
0,0 0,0
0,0
0,46
0,01 0,0
0,0
0,21
0,02 0,0
0,0
0,0
0,0 0,01
0,0
0,08
0,0 0,0
0,0
0,04
0,0 0,0
0,0
0,0
0,0 0,03
0,0
0,26
0,0 0,0
0,0
0,0
0,1 0,0
0,0
0,0
0,02 0,0
0,0
0,0
0,0 0,0
0,0

Challenges

- ➔ Data complexity
- ➔ Curiosity mode: When to explore
- ➔ Sparse rewards: No continuous feedback
- ➔ Locomotion: Controlling body and reaching goals



Reference: http://www.cs.princeton.edu/~andyz/pacmanRL